FlexAttention for Efficient High-Resolution Vision-Language Models

Junyan Li[1], Delin Chen[1], Tianle Cai[2], Peihao Chen[3], Yining Hong[4], Zhenfang Chen[5], Yikang Shen[5], and Chuang Gan[1,5]

[1] UMass Amherst
[2] Princeton University
[3] South China University of Technology
[4] University of California, Los Angeles
[5] MIT-IBM Watson AI Lab
{junyanli,delinchen}@umass.edu
tianle.cai@princeton.edu
{phchencs,yninghong,chenzhenfang2013,yikang.shn,ganchuang1990}@gmail.com

Abstract. Current high-resolution vision-language models encode images as high-resolution image tokens and exhaustively take all these tokens to compute attention, which significantly increases the computational cost. To address this problem, we propose FlexAttention, a flexible attention mechanism for efficient high-resolution vision-language models. Specifically, a high-resolution image is encoded both as high-resolution tokens and low-resolution tokens, where only the low-resolution tokens and a few selected high-resolution tokens are utilized to calculate the attention map, which greatly shrinks the computational cost. The high-resolution tokens are selected via a high-resolution selection module which could retrieve tokens of relevant regions based on an input attention map. The selected high-resolution tokens are then concatenated to the low-resolution tokens and text tokens, and input to a hierarchical self-attention layer which produces an attention map that could be used for the next-step high-resolution token selection. The hierarchical self-attention process and high-resolution token selection process are performed iteratively for each attention layer. Experiments on multimodal benchmarks prove that our FlexAttention outperforms existing high-resolution VLMs (*e.g.,* relatively ∼9% in V* Bench, ∼7% in TextVQA), while also significantly reducing the computational cost by nearly 40%.[1]

Keywords: High-resolution Image · Vision-language Model · Attention Mechanism

1 Introduction

Large vision-language models (VLMs), such as those described in [28, 30], exhibit remarkable capabilities across a range of multimodal tasks including image

[1] Project page: https://vis-www.cs.umass.edu/flexattention

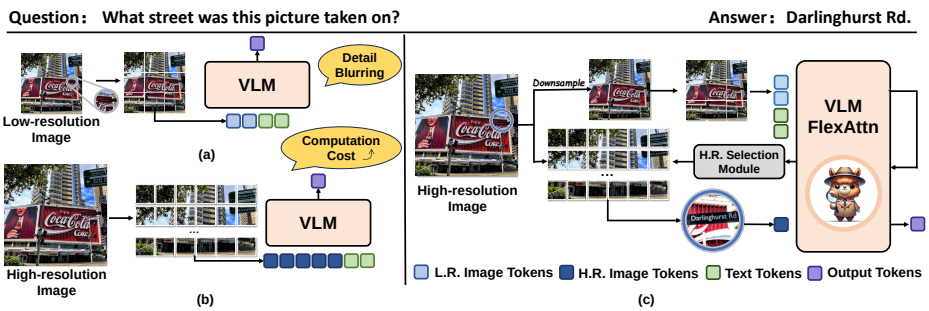

Fig. 1: An overview of VLMs processing high-resolution images for the VQA task. (a) low-resolution VLM will first downsample the high-resolution image to meet its vision encoder requirement. The detail in the low-resolution image is missing, thus it is hard for it to correctly answer the question. **(b)** high-resolution VLM can take the high-resolution image as input, at the cost of a large amount of high-resolution image tokens, leading to excessive computational cost. **(c)** Equipped with our FlexAttention, the model encodes the whole high-resolution image and dynamically selects a small portion of the high-resolution feature that the model is paying attention to during the generation, thus avoiding the high computational cost.

captioning, visual question answering, image-text matching, and so on. However, these models typically process images at relatively low resolutions (*e.g.,* 224 224 or 336 336), thus struggling in scenarios where detailed scrutiny of small regions (e.g., minor texts or small objects) is required. This limitation becomes evident, for instance, in Fig. 1 (a), where these models fail to discern the words on the printed sign due to the constraints of low-resolution inputs.

To address this problem, several high-resolution VLMs (*e.g.*, LLaVA-1.5-HD [35] and CogAgent [20]) have been proposed, which could take high-resolution images as inputs and encode them as high-resolution tokens. Although such models provide a more detailed examination of small regions, they exhaustively process all high-resolution tokens to compute attention, which places a heavy burden upon computational resources. These models deviate from the way human beings perform visual reasoning. Instead of memorizing all pixel-perfect details, we tend to maintain a coarse representation at first, and attend to relevant regions for retrieval of more details only when instilled with external stimuli [5, 41, 41, 42]. It's essential that high-resolution VLMs could also flexibly and dynamically attend to the regions of interest based on low-resolution features for high-resolution detail retrieval.

To this end, we present FlexAttention, a novel attention mechanism that could be seamlessly plugged into most vision-language models to empower their abilities to perceive images with higher resolutions in an efficient manner. Specifically, as is shown in Fig. 1 (c), FlexAttention takes a high-resolution image as input, and encodes the image both as high-resolution image tokens and low-resolution image tokens. For computational efficiency, we only feed the low-resolution tokens and text tokens to the first few layers to roughly understand

the entire image. For subsequent layers, only the low-resolution tokens and a very small portion of high-resolution tokens are utilized to calculate the attention, which significantly shrinks the computational cost. At each decoder layer with FlexAttention, we have a high-resolution feature selection module and a hierarchical self-attention module. The high-resolution feature selection module retrieves high-resolution tokens of relevant regions based on the input attention map. The selected high-resolution tokens are concatenated to the low-resolution tokens along with text tokens, and input to the hierarchical self-attention module. The hierarchical self-attention module produces an attention map, which could be used for the high-resolution token selection that selects high-resolution tokens that are input to the next-layer hierarchical self-attention. The two modules are iteratively processed until the last layer, which produces the final answer through a projector.

We evaluate our FlexAttention on several high-resolution multimodal benchmarks, including general benchmarks such as V* Bench [53] and Magnifierbench [27], as well as domain-specific benchmarks such as TextVQA [45] for text understanding and RSVQA [38] for remote sensing. We show a better performance than other high-resolution methods with nearly 40% computational cost reduction, proving the efficiency of our method. What's more, we achieve a higher score in V* Bench compared to commercial chatbots such as GPT-4V [1].

2 Related Works

Vision Language Models. Our work is closely related to the research that tried to train large multimodal models [33,39,46,47,60,61] for various vision language tasks like visual question answering [22, 45], referring expression comprehension [23, 57] and text-based image retrieval [32, 56]. Traditional methods [19, 29] usually collected large vision-language datasets and learned joint representation between vision and language from scratch to handle different tasks. Such models usually worked well in in-domain data but performed inferior in the benchmarks that require common sense understanding and outside world knowledge [27, 40].

Later, large language models (LLMs) [6, 49, 50] showed impressive power in natural language understanding and reasoning, which brought new possibilities and capabilities to the research of vision and language. A series of large vision-language models have been proposed, which typically connect a pre-trained vision encoder [13,44] with a pre-trained large language model [9,50]. Flamingo [2] first used the cross-attention mechanisms to encode the visual context into the LLMs. BLIP2 [31] proposed the QFormer, which uses a bert model [24] to transform the visual features into a set of learned tokens for LLMs. Fuyu [4] directly projected the image patches into inputs for LLMs to get rid of pre-trained vision encoders. While these models have impressive performance on commonsense understanding and perform incredibly well on traditional vision-language tasks, they often fail to handle tasks that require high-resolution inputs [27, 53] due to two reasons: 1) most VLMs utilize CLIP [44] as their vision encoder, and this limits the input image size of these VLMs to the fixed and relatively small resolu-

tion that CLIP is trained on (e.g., 224x224), and 2) they lack model mechanisms to efficiently handle long image patch sequences, which will lead to the excessive computational cost when the number of image patches increased quadratically with the image resolution increased.

High-Resolution VLMs. To improve VLMs' capability to handle inputs with high resolutions, several VLMs have been proposed [14, 20, 36]. DocPedia [14] transformed the image into a frequency domain to maintain better semantics of the high-resolution images. LLaVA 1.6 [36] designed inputs of various scales to meet the needs of different tasks and balance efficiency and performance. While these models relieved the problem of dense computation, they are orthogonal to our method and have not designed any new attention mechanisms to handle the quadratic computational cost increase challenge introduced by the self-attention mechanism. Recently, CogAgent [20] designed a new vision encoder for high-resolution image input. Different from us, it requires calculating the dense correspondence between the hidden states and the whole high-resolution image feature through cross-attention at every layer of the large language model, making it less efficient. Also, the data for training the model is not publicly available while we are planning to release all data, code, and models for the whole research community.

E□cient Mechanisms for Sequence Modeling. Our work relates to the development of efficient mechanisms for sequence modeling. One approach tackles the quadratic complexity of standard attention mechanisms concerning sequence length. This is done by using structured approximations [10, 25, 43, 48, 52, 59] or linear attention [7]. Another approach replaces attention entirely with recurrent-style models, such as Recurrent Neural Networks (RNNs) and state-space models [17, 18, 21, 54]. Of particular relevance is the work by Yang *et al*. [55], who introduced a hierarchical attention network for document classification. Their model uses a hierarchical structure and two-level attention mechanisms to improve document representation. Our work diverges by focusing on efficient mechanisms specifically designed for high-resolution image inputs, ensuring seamless cooperation with the computations of large language models.

3 Preliminary

Notation. We define some terms that will be used throughout the paper. For a high-resolution vision-language model, we define its high-resolution image input as I_{HR} and the text input as T. Furthermore, we define the low-resolution image tokens as f_{LR}, the high-resolution image tokens as f_{HR}, and the text tokens as f_T. The hidden state for the VLM is denoted as $H \in \mathbb{R}^{N \times D}$, with a sequence length of N and a hidden state size of D. The hidden state H comprises N_i low-resolution image tokens followed by N_t text tokens. We define f_{SHR} as the selected subset of M high-resolution image tokens f_{HR}.

Autoregressive Large Language Models. Autoregressive large language models (LLMs) such as LLaMA [50] play a crucial role in most vision-language models as they are responsible for taking both image and text tokens as input

and generating the answer sequence. An autoregressive LLM is constituted by several stacked decoder layers. Each decoder layer has two sub-layers. The first is a self-attention module, and the second is a feed-forward (FFN) layer. A skip connection is employed around each of the two sub-layers, followed by layer normalization (LN). In short, the output of each sub-layer is $\mathrm{LN}(x + \mathrm{SubLayer}(x))$. For simplicity, layer normalization will be omitted in the subsequent discussion.

Self-attention and Attention Map. Self-attention [51] is the basic module for a decoder layer. For the self-attention, given input hidden state $H \in \mathbb{R}^{N \times D}$, it will first utilize a linear projection layer to project H into Q, K, and V, namely the query, key, and value matrix, and performs the following calculation:

$$\text{Self-attention}(H) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V; \tag{1}$$

where $Q = HW_Q$, $K = HW_K$, $V = HW_V$ and $W_Q/W_K/W_V \in \mathbb{R}^{D \times d}$ is the learnable linear projection matrix. Specifically, the attention map Map is obtained after the softmax operation:

$$Map = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) : \tag{2}$$

The attention map Map is an NxN matrix that measures the importance between tokens: the (i, j) attention value in the attention map indicates the importance of the j-th token to the i-th token, and a higher value means that the j-th token is more important to the i-th token.

Limitation of Self-attention. The computational cost of the self-attention mechanism is characterized by a quadratic increase relative to the sequence length N of the hidden state H. This computational complexity is further amplified when integrating high-resolution images, as it substantially increases the number of image tokens, consequently extending the length of the hidden state. As a result, the computational requirements of the self-attention mechanism undergo a significant escalation, making the processing of high-resolution image inputs impractical due to the prohibitive computational overhead.

4 Vision-language Model with FlexAttention

4.1 Overall Architecture

To solve the limitations of self-attention when dealing with high-resolution images, we introduce **FlexAttention**, which efficiently analyzes high-resolution images by dynamically attending to important regions of high-resolution images. The **FlexAttention** can be plugged into most vision-language models by replacing their self-attention module with our proposed **FlexAttention** module.

As shown in Fig. 2, the modified vision-language model consists of $N_{SA} + N_{FA}$ decoder layers, where the first N_{SA} layers are with the vanilla self-attention

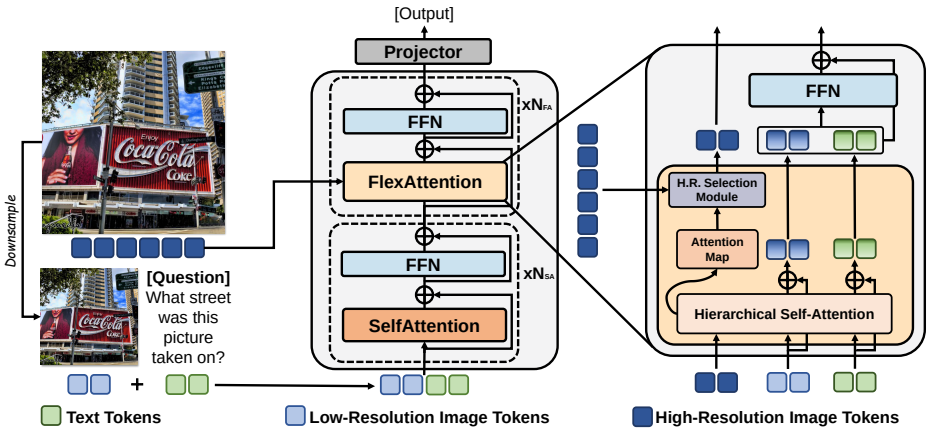

Fig. 2: An Overview of FlexAttention. Within each FlexAttention layer, the encoded high-resolution image features are selected according to the input attention map. These selected features are then inputted into the hierarchical self-attention mechanism alongside input hidden states, which encompass both low-resolution image tokens and text tokens, for computation.

module and the last N_{FA} layers are with our proposed FlexAttention module. Given a high-resolution image, we first downsample it to a low-resolution one and feed both images into an image encoder to get high-resolution and low-resolution image tokens, respectively. For computational efficiency, we only feed the low-resolution image tokens and text tokens to the first N_{SA} layers to roughly understand the whole image. For the subsequent N_{FA} decoder layers with FlexAttention, to efficiently perceive more image details, we additionally feed it with selected high-resolution image tokens. Specifically, FlexAttention consists of two modules: a high-resolution feature selection module and a hierarchical self-attention module. Instead of feeding forward all high-resolution tokens, the high-resolution feature selection module flexibly selects important tokens for the next layer according to an attention map. The hierarchical self-attention module is designed to fuse the selected high-resolution information into the original hidden state. Finally, we use a projector linear layer to produce textual output.

4.2 High-resolution Feature Selection Module

For an autoregressive LLM, the next token is predicted by the last hidden state of the last token. By inspecting the attention values of all other tokens corresponding to the last token in the attention map in Eq. 2, we can find out which tokens the model is paying attention to when generating the next predicted token. When it comes to the vision-language model, this also applies to image tokens f_{LR}. Those image tokens that possess a high attention value can be treated as relevant to important image regions when generating the next token. Although the details contained in the low-resolution image tokens are limited,

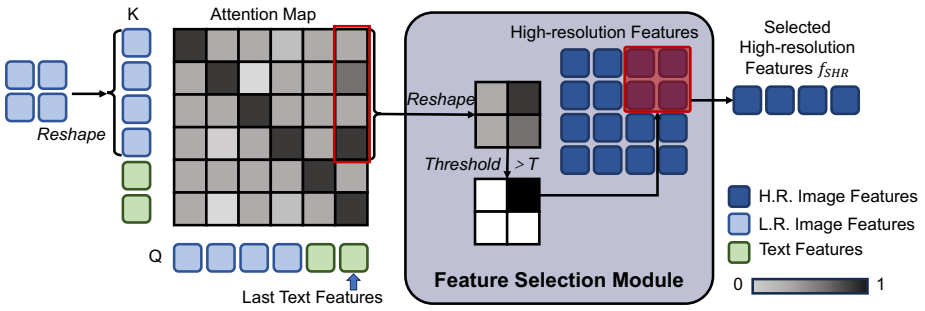

Fig. 3: Illustration of high-resolution feature selection module.

we could retrieve the high-resolution details of the same image regions that have been attended to. Therefore, instead of feeding all high-resolution tokens to the attention module which will lead to excessive computational cost, we dynamically select a very small portion (approximately 10%) of the high-resolution tokens, namely f_{SHR}, and only forward this portion to the attention module.

As is shown in Fig. 3, we take the first N_i values from the last column of the attention map, which corresponds to the importance of the low-resolution image tokens to the last text token, and reshape this 1-D vector to a 2-D map, denoted as the attention mask. Each value in this mask is linked with a patch in the low-resolution image I_{LR}, indicating that patch's importance. The mask is normalized, binarized, and then resized to the same size as the high-resolution feature patch tokens to form the high-resolution selection mask, which serves as the selection decision on whether to select the token of a patch or not. Finally, we apply this mask to the high-resolution image tokens to get the selected high-resolution feature f_{SHR}.

4.3 Hierarchical Self-attention Module

The hierarchical self-attention is the core mechanism to fuse the information from the selected high-resolution tokens f_{SHR} into the hidden state H which consists of both low-resolution tokens and the text tokens. It takes the selected high-resolution tokens $f_{SHR} \in \mathbb{R}^{M \times D}$ and the hidden state $H \in \mathbb{R}^{N \times D}$ as inputs, and outputs the attention map Map' and the updated hidden state H'. The calculation of the hierarchical self-attention is summarized as

$$Q = HW_Q; \tag{3}$$

$$K_{all} = Concat(HW_K; f_{SHR}W'_K); \tag{4}$$

$$V_{all} = Concat(HW_V; f_{SHR}W'_V); \tag{5}$$

$$\text{Hierarchical Self-attention}(H; f_{SHR}) = \text{softmax}\left(\frac{QK_{all}^T}{\sqrt{d_k}}\right)V_{all}; \tag{6}$$

Algorithm 1 Inference Algorithm of VLM with FlexAttention.

Input: High-resolution Image I_{HR}, Text T,
Sub-modules: Image Encoder $E_i(\cdot)$, Text Tokenizer $E_t(\cdot)$, Self-Attention $A(\cdot)$, Hierarchical Self-Attention $HA(\cdot)$, Feed-Forward Network $FFN(\cdot)$, Prediction Head $Head(\cdot)$
Parameters: #Self-Attention layer N_{SA}, #FlexAttention layer N_{FA}
Downsample I_{HR} to low-resolution image I_{LR}.
Generate image and text tokens $f_{HR} = E_i(I_{HR})$, $f_{LR}^0 = E_i(I_{LR})$, $f_T^0 = E_t(T)$
$H^0 = Concat(f_{LR}^0, f_T^0)$
# Decoder layers with self-attention
for i $= 1 \ldots N_{SA}$ **do**
    $Map^i, H^i = A(H^{i-1})$ # Self-attention
    $H^i = H^i + H^{i-1}$ # Skip connection
    $H^i = FFN(H^i) + H^i$ # FFN + skip connection
end for
# Decoder layers with FlexAttention
for i $= N_{SA}{+}1 \ldots N_{SA}{+}N_{FA}$ **do**
    $f_{SHR}^{i-1} = R(f_{HR}, Map^{i-1})$ # Select attended high-resolution feature
    $Map^i, H^i = HA(H^{i-1}, f_{SHR}^{i-1})$ # Hierarchical attention
    $H^i = H^i + H^{i-1}$ # Skip connection
    $H^i = FFN(H^i) + H^i$ # FFN + skip connection
end for
Generate output tokens from $Head(H^{N_{SA}+N_{FA}})$

where $W_Q/W_K/W_V/W_K^0/W_V^0 \in \mathbb{R}^{D \times d}$ is the learnable linear projection matrix. $K_{all} \in \mathbb{R}^{(N+M) \times d}$ and $V_{all} \in \mathbb{R}^{(N+M) \times d}$ are the key and value matrix that fuses the information from high-resolution features. Similar to self-attention, we can obtain an attention map after the softmax operation:

$$Map^0 = \text{softmax}\left(\frac{QK_{all}^T}{\sqrt{d_k}}\right) : \tag{7}$$

Different from self-attention, this attention map Map^0 has a shape of $N \times (N+M)$ as it additionally contains the attention values of high-resolution tokens corresponding to other tokens. We only keep the first $N \times N$ attention values of the matrix shown in Eq. 7 to be the attention map Map used to select the high-resolution feature that will be used in the next layer. A pseudo algorithm for how the vision-language model with our FlexAttention works is described in Alg. 1.

4.4 Complexity Analysis

FlexAttention offers the advantage of executing computations akin to traditional self-attention, thereby minimizing alterations to the model's architecture while facilitating an efficient fusion of multi-grained features. Let the length of the selected high-resolution feature be M, the length of the original hidden state

be N, and the hidden state size be D. The computational complexity of our hierarchical self-attention is

$$\mathsf{T} = \mathbf{O}((M + N)ND): \tag{8}$$

If not using our hierarchical self-attention and directly adding the high-resolution image along with the low-resolution one, the computational complexity will be

$$\mathsf{T}_{original} = \mathbf{O}((M + N)^2D): \tag{9}$$

For vanilla self-attention, the addition of an extra high-resolution feature will lead to a quadratic increase in computation time due to the need to process a significantly larger matrix, as every additional element in the sequence adds to the computational load on a per-element basis. However, the hierarchical self-attention mechanism employed by FlexAttention cleverly mitigates this issue by maintaining a linear relationship in terms of the addition of high-resolution features, thereby considerably reducing the computational burden.

5 Experiments

We evaluate FlexAttention on both high-resolution multimodal benchmarks [27, 38, 45, 53] and general multimodal benchmarks [3, 15, 22, 34, 37, 57, 58], comparing our method with the low-resolution large vision-language models [8, 11, 28] as well as other high-resolution methods [20, 35].

5.1 Implementation

To assess the performance and efficiency of our proposed FlexAttention, we integrated it into LLaVA-1.5-7b [35], resulting in a variant we call LLaVA-FlexAttn. The input resolution is set to be 1008x1008, which is three times the original input image resolution. We then compared this variant with the original LLaVA-1.5-7b model to demonstrate the advantages of utilizing high-resolution image inputs. We also compare FlexAttention with the methods used in LLaVA-1.5-HD [35] and CogAgent [20] that enables the input of high-resolution image in those models, to show the efficiency of our proposed method.

LLaVA-1.5-HD [35] In this model, the high-resolution image tokens act like normal tokens. They are concatenated with the low-resolution image tokens and are fed into the large language model together. Since this model has not been publicly released yet, we re-implement it on top of the codebase for LLaVA-1.5. We use the LLaVA-1.5-7b model as the base model. The input resolution of the high-resolution image is set to 448x448 following the setting in [35]. We refer to this baseline as LLaVA-HD.

CogAgent [20] In this model, the high-resolution feature is perceived using a cross-attention module. In the cross-attention module, the high-resolution features serve as the key and value, while the hidden states, comprising both low-resolution image tokens and text tokens, act as the query. Since CogAgent is trained on document and GUI style data, and the data processing and training code has not been released, for fair comparison on the effectiveness of the high-resolution operator used in CogAgent, we transfer the cross-attention module in CogAgent's inference codebase to LLaVA-1.5 and re-implement the training code. We use the LLaVA-1.5-7b model as the base model. The input resolution of the high-resolution image is set to 1008x1008 to keep it the same as ours. We refer to this baseline as LLaVA-XAttn.

5.2 Training Settings

For a fair comparison, both high-resolution baselines (LLaVA-HD and LLaVA-XAttn) and our LLaVA-FlexAttn load the pre-trained weight for LLaVA-1.5-7b as initialization, and are then finetuned on the LLaVA-1.5-7b's finetuning dataset for one epoch. We use a batch size of 1152 and a learning rate of 2e-5, with a cosine learning rate scheduler. All evaluations are performed in a zero-shot manner.

5.3 Evaluation on High-resolution Multimodal Benchmarks

Datasets. We conduct experiments on four high-resolution benchmarks: V* Bench [53], MagnifierBench [27], TextVQA [45] and RSVQA-HRBEN [38]. The first two benchmarks focus on evaluating the model's capability on general high-resolution VQA, while the last two benchmarks focus on evaluating the model's performance on domain-specific high-resolution VQA such as TextVQA for text understanding and RSVQA-HRBEN for remote sensing.

Baselines. We conduct a comparative analysis between LLaVA-FlexAttn and two categories of Vision-Language Models (VLMs): low-resolution VLMs, specifically InstructBLIP [11], Otter [28], MiniGPT-4 [62], MiniGPTv2 [8] and LLaVA [35], as well as high-resolution VLMs that were re-implemented for this research. Additionally, comparisons are made with commercial chatbots such as GPT-4V [1], and specialist VLM such as GeoChat [26], to evaluate the significance of high-resolution image input capabilities.

Results. Table 1 shows the evaluation results on the two high-resolution general VQA benchmarks. In general, all three high-resolution VLMs are better than low-resolution VLMs, while our model is consistently better than other high-resolution VLMs, with an overall accuracy of 54.5% for V* Bench and an accuracy of 35.0% for MagnifierBench. Compared to the base model LLaVA-1.5-7b, the overall accuracy gain for V* Bench is 6.9% and the accuracy gain for MagnifierBench is 8.2%. Compared to other high-resolution methods, our method achieves comparable and even higher accuracy at the cost of much lower TFLOPs than other high-resolution methods, nearly 30% lower TFLOPs than LLaVA-HD (from 24.9 to 17.1) and over 37% lower TFLOPs than LLaVA-XAttn

	Resolution	V* Bench			MagnifierBench
		Attribute	Spatial	Overall	
Commercial Chatbots					
Bard [16]	-	31.3	46.1	37.2	-
Gemini Pro [12]	-	40.9	59.2	48.2	-
GPT-4V [1]	-	**51.3**	60.5	**55.0**	-
Low-resolution VLMs					
InstructBLIP [11]	224^2	25.2	47.4	34.0	5.6
Otter [28]	224^2	27.0	56.6	38.7	25.7
MiniGPT-4 [62]	224^2	30.4	50.0	38.2	22.6
LLaVA-1.5-7b [35]	336^2	41.7	56.6	47.6	26.8
High-resolution VLMs					
LLaVA-HD [35]	448^2	45.2	<u>61.8</u>	51.8	**35.0**
LLaVA-XAttn [20]	1008^2	42.6	56.6	48.2	32.2
LLaVA-FlexAttn	1008^2	<u>47.8</u>	**64.5**	<u>54.5</u>	**35.0**

Table 1: General high-resolution VQA benchmark results comparison.

(from 27.1 to 17.1). Detailed discussion on the TFLOPs and inference time can be found in Sec. 5.6. Thanks to the high-resolution feature selection and hierarchical self-attention, our method can enable the input image resolution to increase three times compared to the original resolution, with the cost of a sub-linear computational cost increasing, achieving a better trade-off between computational cost and accuracy. Compared with GPT-4V on V* Bench, our method shows competitive performance, achieving even higher accuracy on spatial category than GPT-4V, and a comparable overall performance with GPT-4V.

Table 2 presents the results on two high-resolution domain-specific VQA benchmarks. Our LLaVA-FlexAttn is consistently superior to the base model and other high-resolution methods on both RSVQA-HRBEN and TextVQA. Furthermore, our approach surpasses GeoChat [26] in terms of overall accuracy on the RSVQA-HRBEN benchmark, a model explicitly crafted and fine-tuned for remote sensing Visual Question Answering benchmarks. This outcome underscores the efficacy of incorporating high-resolution image inputs, suggesting that the increased detail and clarity provided by high-resolution inputs can significantly improve the model's understanding and processing of intricate visual patterns in specialized VQA tasks.

5.4 Evaluation on General Multimodal Benchmarks

Datasets and Baseline. We evaluate the general vision-language model performance on several multimodal tasks including GQA [22], VQAv2 [3], POPE [34], RefCOCO [57], MM-Bench [37], MME [15], and MM-Vet [58]. This collection

	RSVQA-HRBEN			TextVQA
	Presence	Comparison	Overall	
Low-resolution VLMs				
GeoChat [26]	58.5	83.2	72.3	-
MiniGPTv2 [8]	40.8	50.9	46.5	27.5
LLaVA-1.5-7b [35]	69.8	67.3	68.4	46.0
High-resolution VLMs				
LLaVA-HD [35]	69.0	67.6	68.4	45.6
LLaVA-XAttn [20]	71.4	70.9	71.1	45.5
LLaVA-FlexAttn	**72.2**	**73.1**	**72.7**	**48.9**

Table 2: Domain-specific high-resolution VQA benchmark results comparison.

	RefCOCO	POPE	GQA	VQAv2	MM-Bench	MME	MM-Vet
LLaVA-1.5-7b [35]	75.8	85.9	62.0	78.5	64.3	**1511**	**31.1**
LLaVA-FlexAttn	**79.3**	**85.9**	**62.2**	**78.7**	**65.7**	1479	29.4

Table 3: Comparison of the multimodal capability between the base model and our model on a broad range of multimodal benchmarks.

of benchmarks assesses the model's overall capabilities, including spatial understanding, localization, ability to avoid hallucinations, and performance in academic-oriented tasks. We compare our method to the base model LLaVA-1.5-7b to analyze the change in the model's general ability.

Results. In Table 3 we show that with our FlexAttention, the performance on RefCOCO is improved. RefCOCO requires the localization of an object based on a referring expression. Thus, incorporating a high-resolution feature could reduce the challenge of identifying a small object and enhance the precision of its location prediction. We achieve a similar rate of hallucination on POPE and maintain similar performance on large-scale VQA benchmarks. This indicates that incorporating FlexAttention does not impact the model's overall capability.

5.5 Ablation Study

H.R. Feature Selection Strategy. We first conduct an ablation study to verify the effectiveness of the key design of our method, which is the strategy to select high-resolution features using the attention map. We compare our attention map selection strategy with two naive baseline strategies: 1) random selection, which means randomly selecting a few patches of the high-resolution features, and 2) center selection, which means selecting the center region of the high-resolution features. The selection ratio is kept to approximately the same as our attention map selection strategy which is about 10%. We finetune them respectively using

	Magnifierbench	TextVQA
Random	31.4	44.5
Center	30.7	45.9
Attn. Map	**35.0**	**48.9**

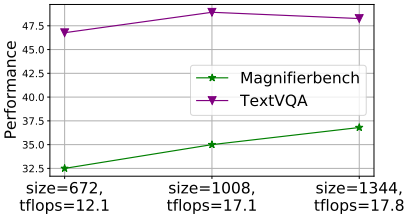

Fig. 4: Ablation studies of selection strategies (left) and image sizes (right).

RefCOCO Val Acc.	Large	Small	Overall
LLaVA	75.9	41.3	75.4
LLaVA-FlexAttn	78.8 (+2.9)	51.3 **(+10.0)**	78.4 (+3.0)

Table 4: Analysis on accuracy across different object sizes.

the same finetuning dataset and following the same training setting and evaluate their performance on Magnifierbench and TextVQA.

The experiment results in Fig. 4 (left) shows that our attention map selection strategy is better than the other two baseline strategies. For the baseline strategies, since the model cannot dynamically pay more attention to the region that needs to be focused, the benefit of the high-resolution image is limited, and no consistent improvement is observed especially for TextVQA which requires the model to focus on a specific region to give the correct answer.

Impact of Resolution. We also explore the effect of the high-resolution image size. The default setting is 1008x1008, tripling the resolution of the original low-resolution image. Additionally, we introduce two other settings: 672x672 and 1344x1344, doubling and quadrupling the original resolution, respectively. We finetune them respectively using the same finetuning dataset and following the same training setting. We measure their average TFLOPs on Magnifierbench benchmarks and evaluate their performance on Magnifierbench and TextVQA.

Fig. 4 (right) shows the experiment results. We can see that as the resolution increases, the performance on Magnifierbench also increases. Performance on TextVQA significantly enhances when the resolution is increased from 672 to 1008 but sees no further improvement from 1008 to 1344. Since the average resolution of images in TextVQA is 950 811, further increasing the resolution beyond its original resolution is unbeneficial. This pattern is aligned with what we observe for general VQA benchmarks.

Impact of Object Size. For general benchmarks evaluated in Section 5.4, most questions do not focus on small details, and thus cannot reveal the capability of our model for handling high-resolution image. To better evaluate our model on general benchmarks, we divide the benchmark into two subsets according to the size of question-relevant objects, categorizing those larger than 5% of the image

as large objects and the rest as small. We conduct experiments on RefCOCO val set as it provides object sizes.

Table 4 shows that the accuracy improvement on small objects is much higher than large objects. It indicates that even for non high-resolution benchmarks such as RefCOCO, our method can still improve the accuracy when the questions involve small objects or detailed information.

5.6 Inference Time on Hardware

We measure the inference time on hardware to assess the efficiency of our FlexAttention. Models are implemented in PyTorch and the inference time is measured on a single NVIDIA V100 32G GPU. We measure the average TFLOPs and total inference time on two benchmarks: Magnifierbench, in which the model answer is a single letter, and TextVQA, in which the model answer is a short phrase. Warm-up before inference and CUDA synchronization are employed to ensure the accuracy of the measurement results.

The measurement results are presented in Table 5. In Magnifierbench, the inference time reduction is linearly proportional to the theoretical computational cost reduction measure in TFLOPs, and our method is nearly 30% and 40% faster than the two baselines respectively. In TextVQA, the speed superior slightly declined, but still about 15% and 25% faster than baselines. Note that the average output length for TextVQA is longer than Magnifierbench, so the inference time will be affected more by the generation phase, which is memory-bound instead of computation-bound. A discussion is provided in the Supplementary.

	Magnifierbench		TextVQA	
	TFLOPs	Time(s)	TFLOPs	Time(s)
LLaVA-HD [35]	24.9	154	24.5	3273
LLaVA-XAttn [20]	27.1	178	26.7	3741
LLaVA-FlexAttn	**17.1**	**112**	**17.1**	**2839**

Table 5: Average TFLOPs and total inference time measured on NVIDIA V100 GPU.

6 Conclusion

In this paper, we propose FlexAttention, a method designed to enhance large vision-language models by allowing them to efficiently process and derive advantages from high-resolution image inputs. By leveraging dynamic high-resolution feature selection and hierarchical self-attention mechanism, FlexAttention surpasses existing high-resolution methods in terms of performance as well as efficiency. The idea behind FlexAttention can be extended to other long sequence modalities such as video or audio, which can be a crucial future direction.

Acknowledgments

We are grateful to the anonymous reviewers for their valuable feedback. We also extend our thanks to AiMOS for supplying the computational resources necessary for this project.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning (2022)
3. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
4. Bavishi, R., Elsen, E., Hawthorne, C., Nye, M., Odena, A., Somani, A., Taşırlar, S.: Fuyu-8b: A multimodal architecture for ai agents (2024)
5. Broadbent, D.E.: Perception and Communication. Pergamon Press (1958). https://doi.org/10.1037/10037-000
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners (2020)
7. Cai, H., Li, J., Hu, M., Gan, C., Han, S.: Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In: ICCV (2023)
8. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)
9. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (2023), https://lmsys.org/blog/2023-03-30-vicuna/
10. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019)
11. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning (2023)
12. DeepMind, G.: Gemini (December 2023), https://deepmind.google/technologies/gemini
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv (2020)
14. Feng, H., Liu, Q., Liu, H., Zhou, W., Li, H., Huang, C.: Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. arXiv (2023)
15. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023)

16. Google: Bard (Febraury 2023), https://bard.google.com
17. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv (2023)
18. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. ICLR (2022)
19. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: Towards learning a generic agent for vision-and-language navigation via pre-training. In: CVPR (2020)
20. Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Zhang, Y., Li, J., Xu, B., Dong, Y., Ding, M., Tang, J.: Cogagent: A visual language model for gui agents (2023)
21. Hou, H., Yu, F.R.: Rwkv-ts: Beyond traditional recurrent neural network for time series tasks. arXiv preprint arXiv:2401.09093 (2024)
22. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019)
23. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: EMNLP (2014)
24. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
25. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: The efficient transformer. iclr (2020)
26. Kuckreja, K., Danish, M.S., Naseer, M., Das, A., Khan, S., Khan, F.S.: Geochat: Grounded large vision-language model for remote sensing. arXiv preprint arXiv:2311.15826 (2023)
27. Li, B., Zhang, P., Yang, J., Zhang, Y., Pu, F., Liu, Z.: Otterhd: A high-resolution multi-modality model. arXiv (2023)
28. Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., Liu, Z.: Otter: A multi-modal model with in-context instruction tuning. arXiv preprint arXiv:2305.03726 (2023)
29. Li, G., Duan, N., Fang, Y., Gong, M., Jiang, D.: Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In: AAAI (2020)
30. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models (2023)
31. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
32. Li, W., Duan, L., Xu, D., Tsang, I.W.H.: Text-based image retrieval using progressive multi-instance learning. In: ICCV. IEEE (2011)
33. Li, X., Yin, X., Li, C., Hu, X., Zhang, P., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J.: Oscar: Object-semantics aligned pre-training for vision-language tasks. ECCV 2020 (2020)
34. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
35. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning (2023)
36. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/
37. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)
38. Lobry, S., Marcos, D., Murray, J., Tuia, D.: Rsvqa: Visual question answering for remote sensing data. IEEE Transactions on Geoscience and Remote Sensing **58**(12), 8555–8566 (2020)

39. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. NeurIPS (2019)
40. Marino, K., Rastegari, M., Farhadi, A., Mottaghi, R.: Ok-vqa: A visual question answering benchmark requiring external knowledge. In: CVPR (2019)
41. Oishausen, B.A., Anderson, C.H., Van Essen, D.C.: A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. The Journal of Neuroscience **13**(11), 4700–4719 (November 1993)
42. Palmer, S.E.: The psychology of perceptual organization: a transformational approach. In: Beck, J., Hope, B., Rosenfeld, A. (eds.) Human and machine vision. Academic Press, New York (1983), in: Beck, J., Hope, B. & Rosenfeld, A. (Eds.)
43. Peng, H., Pappas, N., Yogatama, D., Schwartz, R., Smith, N.A., Kong, L.: Random feature attention. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), `https://openreview.net/forum?id=QtTKTdVrFBB`
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. PMLR (2021)
45. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8317–8326 (2019)
46. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV (2019)
47. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. arXiv (2019)
48. Tay, Y., Bahri, D., Metzler, D., Juan, D.C., Zhao, Z., Zheng, C.: Synthesizer: Rethinking self-attention in transformer models. arXiv preprint arXiv: 2005.00743 (2020)
49. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv (2023)
50. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
51. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
52. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. arXiv preprint arXiv: Arxiv-2006.04768 (2020)
53. Wu, P., Xie, S.: V*: Guided visual search as a core mechanism in multimodal llms. arXiv preprint arXiv:2312.14135 **17** (2023)
54. Yang, S., Wang, B., Shen, Y., Panda, R., Kim, Y.: Gated linear attention transformers with hardware-efficient training. arXiv preprint arXiv: 2312.06635 (2023)
55. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: NAACL (2016)
56. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. TACL (2014)
57. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: ECCV (2016)

58. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. arXiv preprint arXiv:2308.02490 (2023)
59. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontañón, S., Pham, P., Ravula, A., Wang, Q., Yang, L., Ahmed, A.: Big bird: Transformers for longer sequences. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html
60. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., Gao, J.: Vinvl: Making visual representations matter in vision-language models. CVPR 2021 (2021)
61. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. In: AAAI (2020)
62. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)